Simulation methods

The Monte Carlo Method

Aim: to compute thermal averages of equilibrium systems.

$$\langle A \rangle = \frac{\sum_i \exp(-\epsilon_i/k_B T) A_i}{\sum_i \exp(-\epsilon_i/k_B T)}$$

Where *i* labels all eigenstates of the system, and

$$A_i = \langle i | A | i \rangle$$

Classical limit: replace the SUM over quantum states by an INTEGRAL of phase space

$$\langle A \rangle = \frac{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; A(\mathbf{p}^N, \mathbf{r}^N) \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}$$

Where *H* is the Hamiltonian of the system and $\beta=1/kT$

In replacing the sum by an integral, we have attributed a “volume” h^{3N} to every quantum state

---

Problem:

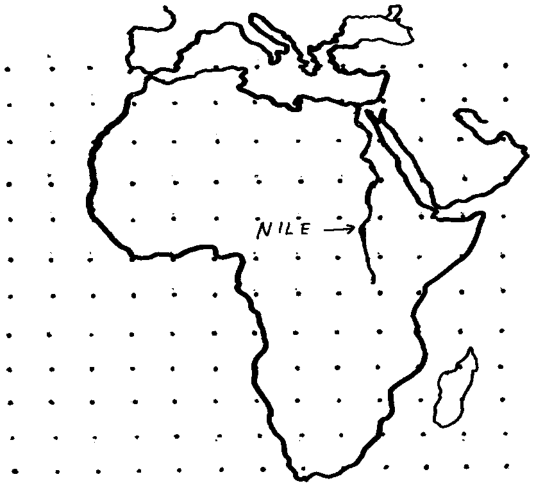
We cannot compute the sum over all quantum states (because there are so many)

And we cannot compute the classical integral either (except the integration over momenta).

Consider “normal” numerical integration

100 particles, 3 dimensions, 10 points in every direction.

Requires 10^{300} points for a very poor estimate of the integral…



Similar problem (but much less serious):

Measure the depth of the Nile by quadrature…

BETTER STRATEGY:

IMPORTANCE SAMPLING

We wish to perform a RANDOM WALK in configuration space, such that

The number of times that each point is visited, is proportional to its Boltzmann weight.

$$n(\mathbf{r}^N) = c\,\exp[-\beta \mathcal{U}(\mathbf{r}^N)]$$

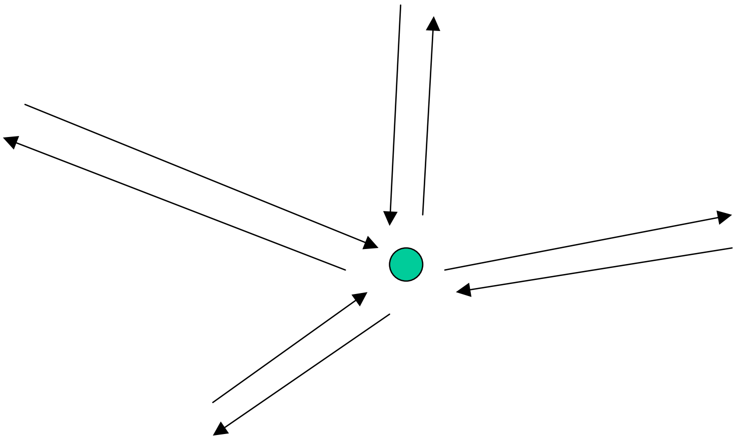
Then

$$\langle A \rangle \approx \frac{1}{L} \sum_{i=1}^{L} n_i\, A(\mathbf{r}_i^N).$$

How do we achieve that?

Whatever our rule is for moving from one point to another, it should not destroy the equilibrium distribution.

That is: in equilibrium we must have



$$\mathcal{N}(o)\sum_n \pi(o \to n) = \sum_n \mathcal{N}(n)\pi(n \to o)$$

Stronger condition:

$$\mathcal{N}(o)\pi(o \to n) = \mathcal{N}(n)\pi(n \to o).$$

For every pair {*n,o*}.

Detailed Balance

Now we construct the transition probabilities

$$\pi(o \rightarrow n) = \alpha(o \rightarrow n) \times \mathrm{acc}(o \rightarrow n)$$

Then, detailed balance implies that:

$$\mathcal{N}(o)\alpha(o \rightarrow n) \times \mathrm{acc}(o \rightarrow n)$$

$$=$$

$$\mathcal{N}(n)\alpha(n \rightarrow o) \times \mathrm{acc}(n \rightarrow o)$$

Often, we choose

$$\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$$

Then it follows that

$$\mathcal{N}(o) \times \mathrm{acc}(o \rightarrow n)$$

$$=$$

$$\mathcal{N}(n) \times \mathrm{acc}(n \rightarrow o)$$

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}$$

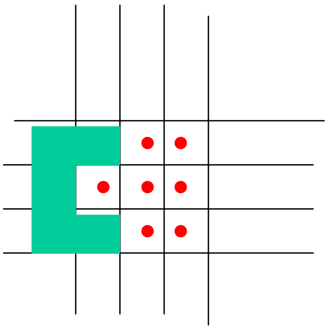
Metropolis, Rosenbluth,Rosenbluth,

Teller and Teller choice:

$$\mathrm{acc}(o \to n) = \min\left(1, \exp\{-\beta[\mathcal{U}(\mathbf{r}'^N) - \mathcal{U}(\mathbf{r}^N)]\}\right)$$

Kirkwood’s objection:

“If a trial move has been rejected, one should not count the original state AGAIN…”



Counter-example:

Ideal gas on a lattice.

More Formal

Transition probability:

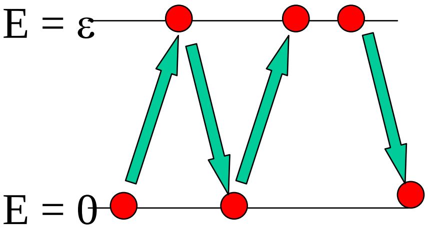
$$\pi(o \to n) = \alpha(o \to n) \times \mathrm{acc}(o \to n)$$

$$\sum_n \pi(o \to n) = 1$$

Probability to accept the old configuration:

$$\pi(o \to o) = 1 - \sum_{n \neq o} \pi(o \to n)$$

Example: a VERY simple MC program



- try to change energy state
- compute $\Delta E = E_{\text{new}} - E_{\text{old}}$
- accept new state if $Ran \leq e^{-\Delta E/kT}$
- reject otherwise
- sample the state of the system
- start again

Algorithm 1 (Basic Metropolis Algorithm)

```
  PROGRAM mc                        basic Metropolis algorithm

  do icycl=1,ncycl                  perform ncycl MC cycles
    call mcmove                     displace a particle
    if (mod(icycl,nsamp).eq.0)
+      call sample                  sample averages
  enddo
  end
```

Comments to this algorithm:

1. *Subroutine* `mcmove` *attempts to displace a randomly selected particle (see Algorithm 2).*
2. *Subroutine* `sample` *samples quantities every* `nsamp`*th cycle.*

Algorithm 2 (Attempt to Displace a Particle)

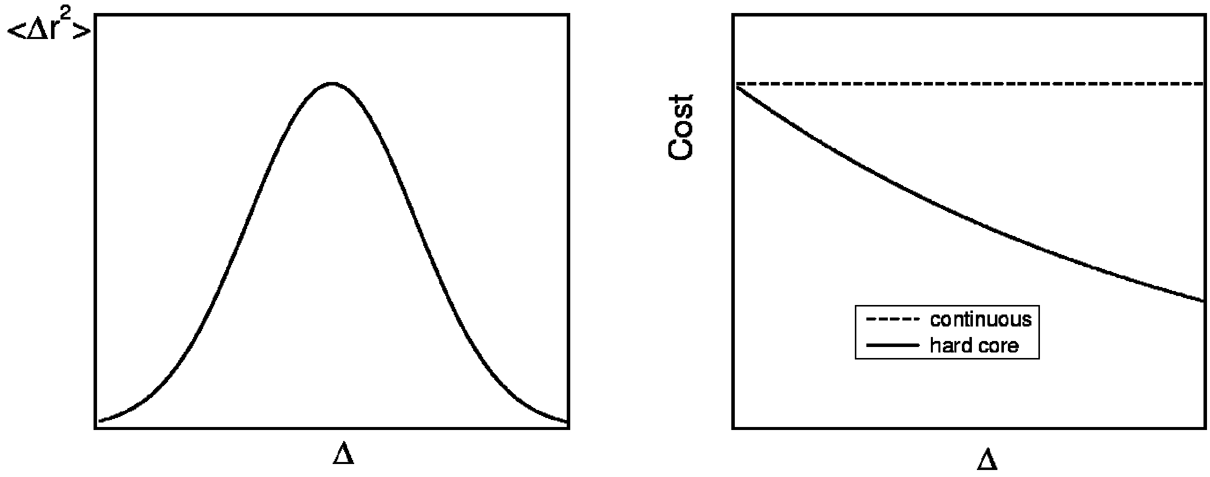
```
  SUBROUTINE mcmove             attempts to displace a particle

  o=int(ranf()*npart)+1         select a particle at random
  call ener(x(o),eno)           energy old configuration
  xn=x(o)+(ranf()-0.5)*delx     give particle random displacement
  call ener(xn,enn)             energy new configuration
  if (ranf().lt.exp(-beta       acceptance rule (3.2.1)
+    *(enn-eno)) x(o)=xn        accepted: replace x(o) by xn
  return
  end
```

Comments to this algorithm:

1. *Subroutine* `ener` *calculates the energy of a particle at the given position.*
2. *Note that, if a configuration is rejected, the old configuration is retained.*
3. *The* `ranf()` *is a random number uniform in* $[0, 1]$.

How “large” should a trial displacement be?



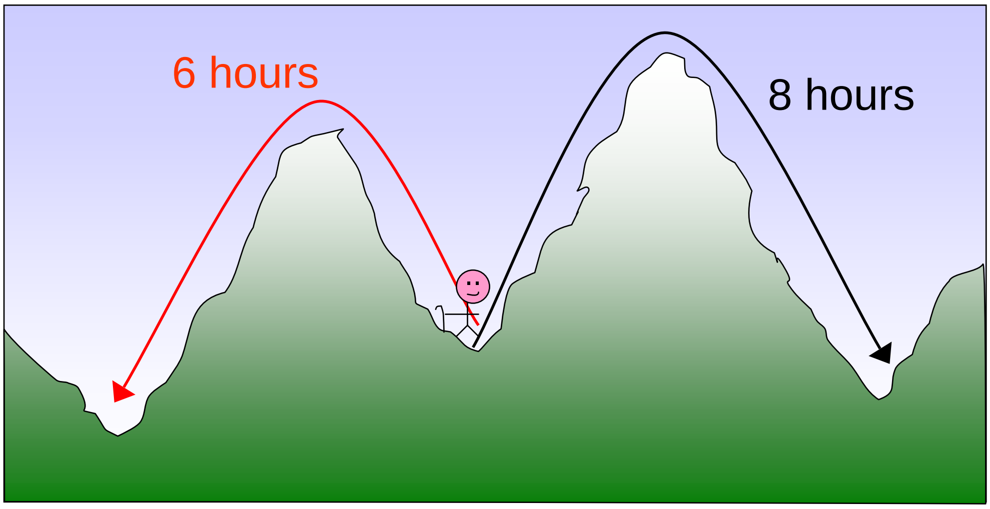
Problem for both MC and MD:

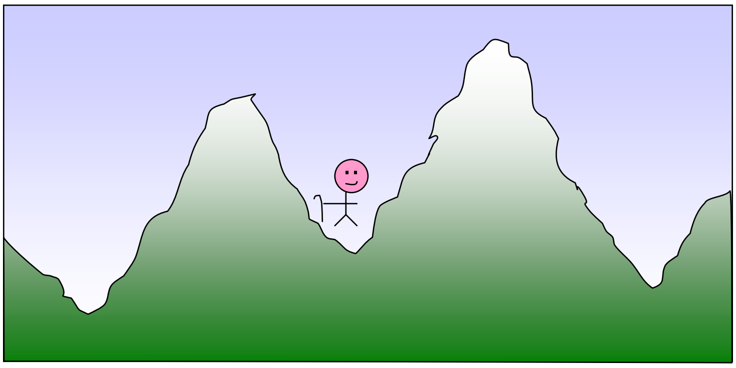
“non-ergodicity”

(i.e. the sampling is limited to a subset of all possible states of the system)

Slow dynamics implies slow equilibration. This is particularly serious for glassy systems.

Mountain hikes

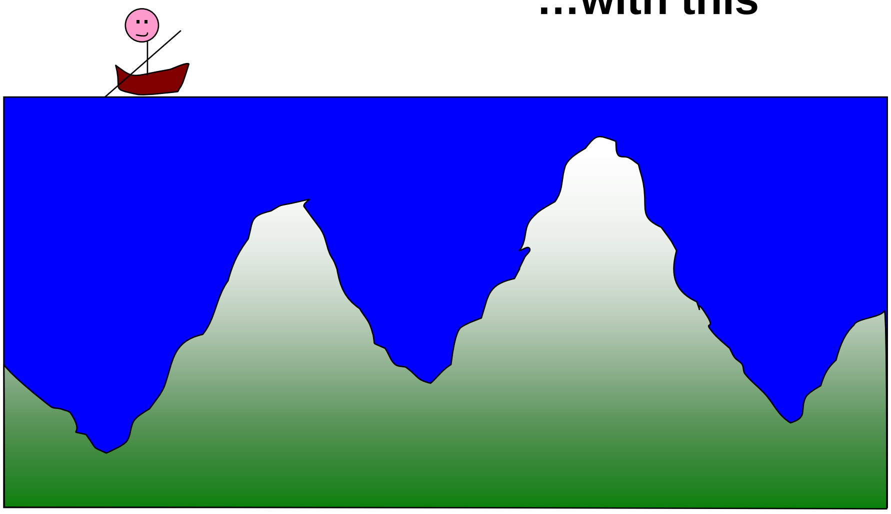




Sampling the valleys

Combine this…

…with this



Parallel Tempering

COMBINE Low-temperature and high-temperature runs in a SINGLE Parallel simulation

---

In practice:

System 1 at temperature T_1

System 2 at temperature T_2

Boltzmann factor

$$\exp(-\beta_1 U_1(r^N))$$

Boltzmann factor

$$\exp(-\beta_2 U_2(r^N))$$

Total Boltzmann factor

$$\exp(-\beta_1 U_1(r^N))\exp(-\beta_2 U_2(r^N))$$

SWAP move

System 1 at temperature T_2

System 2 at temperature T_1

Boltzmann factor

$$\exp(-\beta_2 U_1(r^N))$$

Boltzmann factor

$$\exp(-\beta_1 U_2(r^N))$$

Total Boltzmann factor

$$\exp(-\beta_2 U_1(r^N))\,\exp(-\beta_1 U_2(r^N))$$

Ratio

$$\frac{\text{new Boltzmann weight}}{\text{old Boltzmann weight}}$$

$$=$$

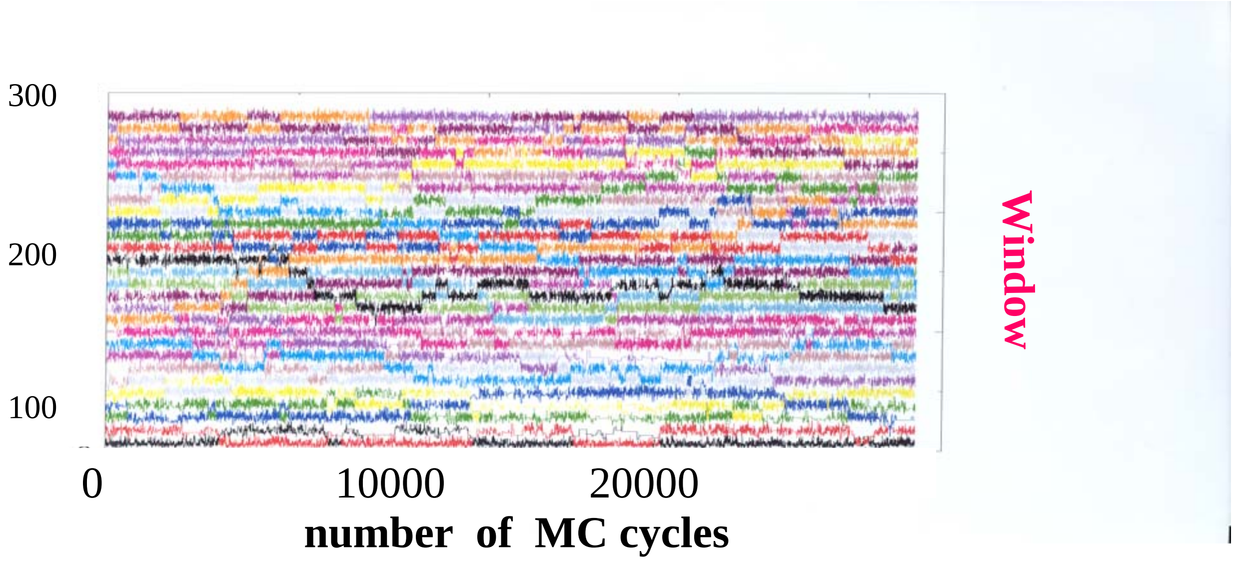
$$\exp\{-(\beta_1-\beta_2)[\mathcal{U}_2(\mathbf{r}^N)-\mathcal{U}_1(\mathbf{r}^N)]$$

Systems may swap temperature if their combined Boltzmann factor allows it.

$$\mathrm{acc(swap)}$$

$$=$$

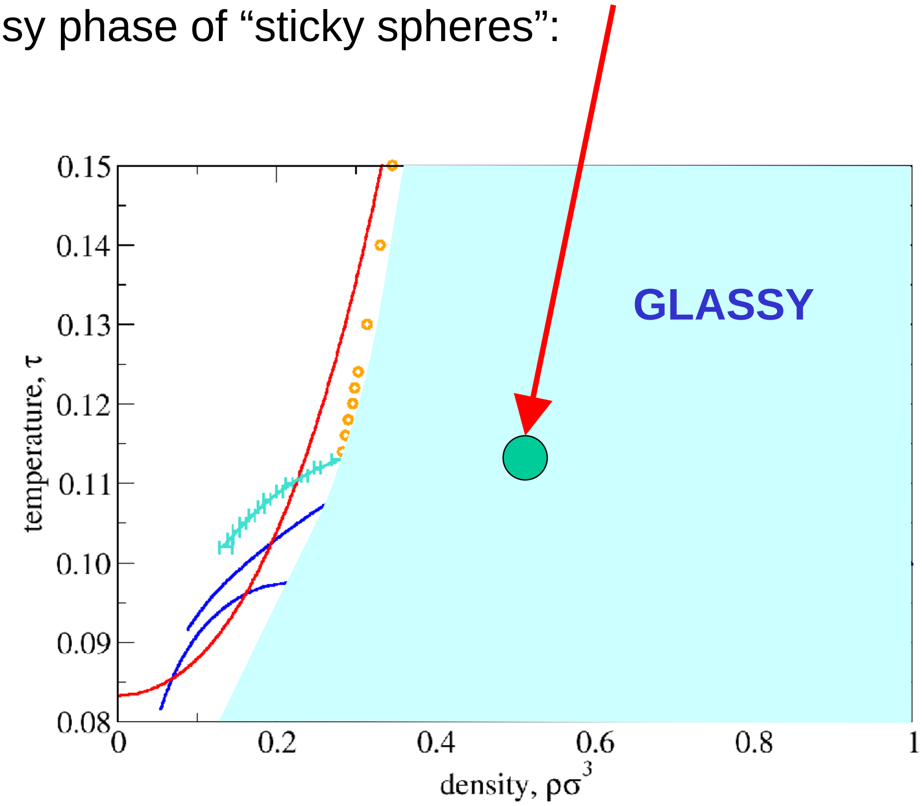
$$\min\left(1, \exp\{-(\beta_1 - \beta_2)[\mathcal{U}_2(\mathbf{r}^N) - \mathcal{U}_1(\mathbf{r}^N)]\}\right)$$



NOTES:

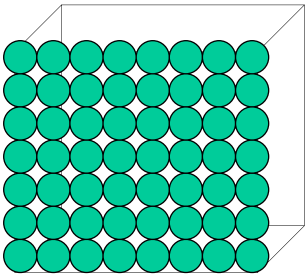
1. One can run MANY systems in parallel
2. The control parameter need not be temperature

Application: computation of a critical point INSIDE the glassy phase of “sticky spheres”:



Practical issues:

1. Boundary conditions
2. Time-saving devices

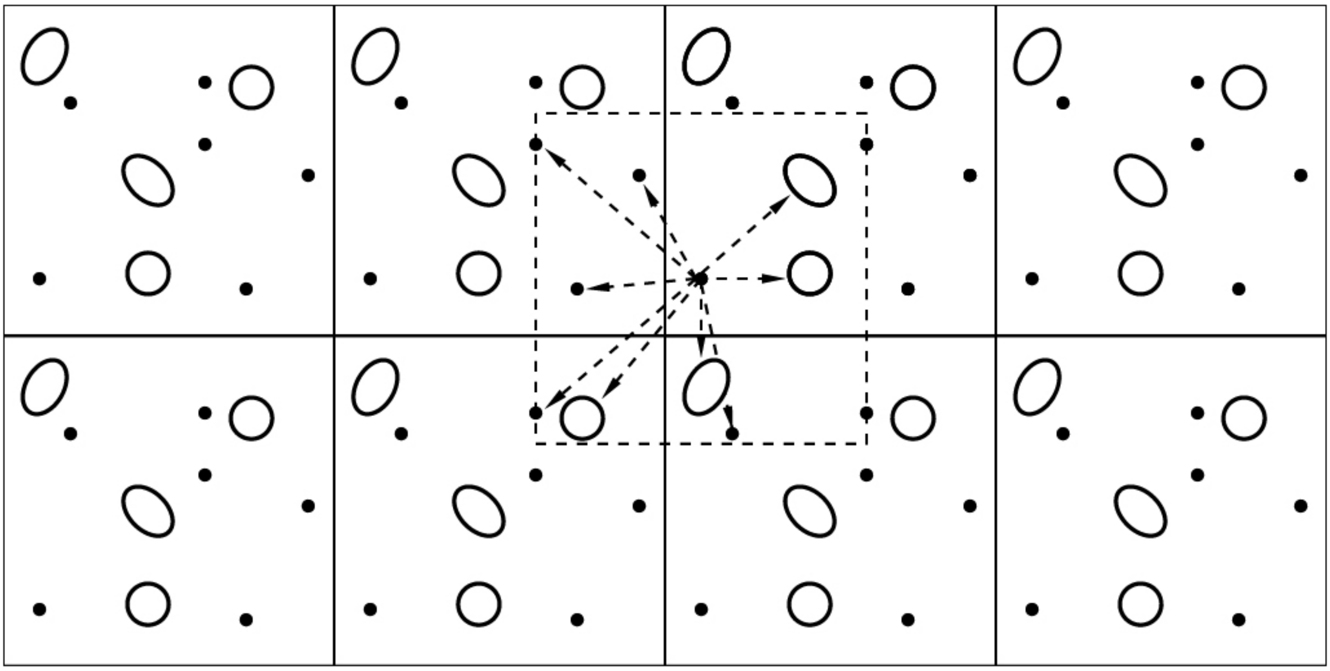


In small systems, boundary effects are always large.

1000 atoms in a simple cubic crystal – 488 boundary atoms.

1000000 atoms in a simple cubic crystal – still 6% boundary atoms…

“Solution” : Periodic boundary conditions



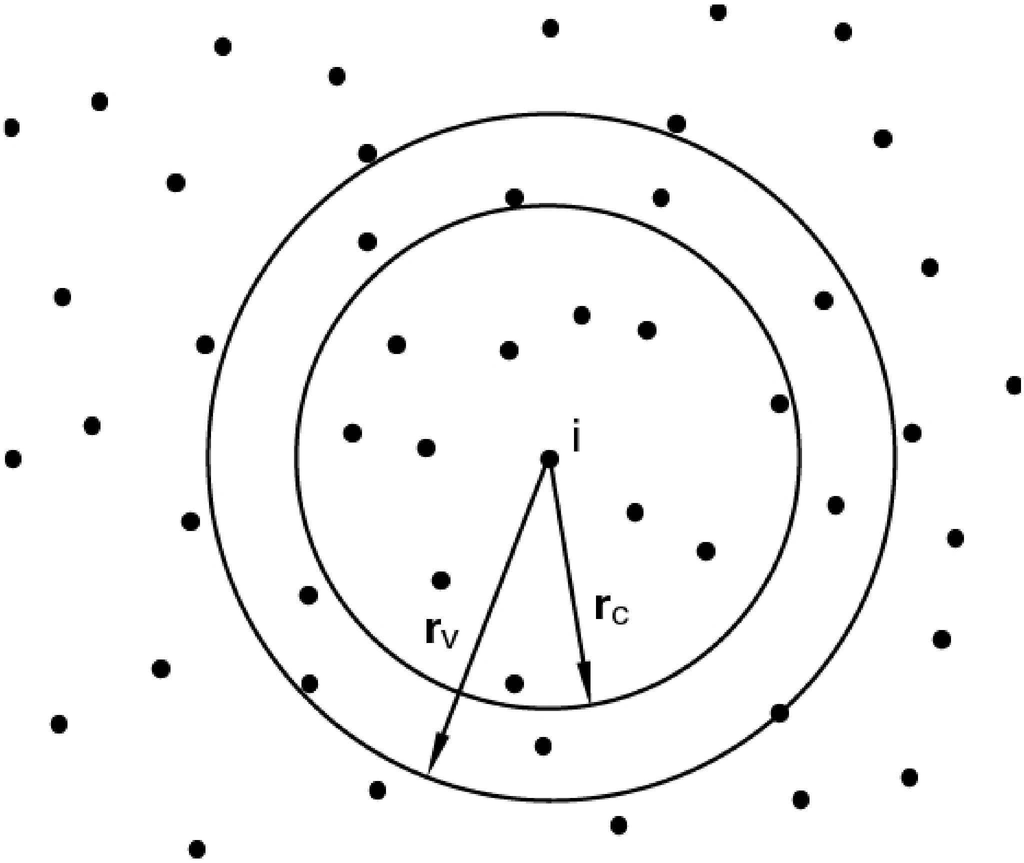
The most time-consuming part of any simulation is the evaulation of all the interactions between the molecules.

In general: $N(N-1)/2 = O(N^2)$

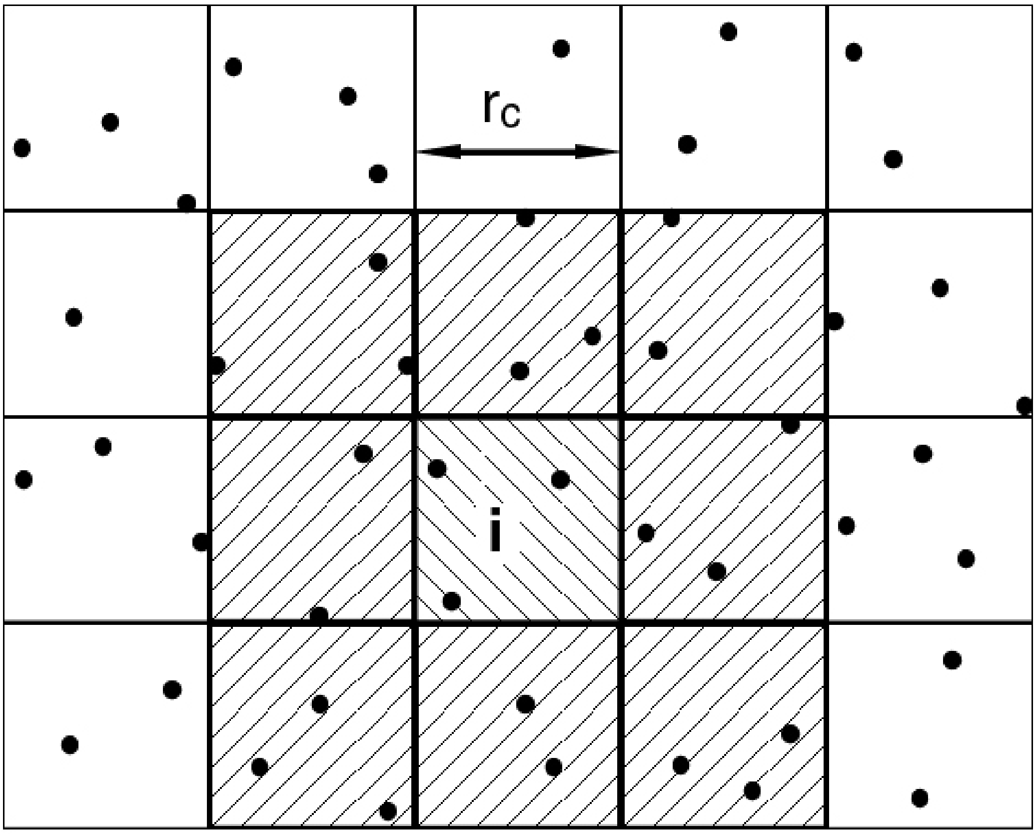
But often, intermolecular forces have a short range:

Therefore, we do not have to consider interactions with far-away atoms…

Verlet list



Link list



The shaky foundations of…

Molecular Dynamics

Molecular Dynamics

The Basis:

$F_i = m_i a_i$

$i = 1, 2, \ldots, N$

N-body problem. Can only be solved numerically

(except in very special cases)

How?

$$X(t+\Delta t) = X(t) + \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2 + \frac{1}{3!}\dddot{X}(t)\Delta t^3 + \ldots$$

...at least, in principle.

Naive approach: truncate Taylor expansion.

$$\xcancel{X(t+\Delta t) \approx X(t) + \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2}$$

ABSOLUTELY FORBIDDEN!

The naive “forward Euler” algorithm

- is not time reversible
- does not conserve volume in phase space
- suffers from energy drift

Better approach: “Verlet” algorithm

$$\mathrm{X}(\mathrm{t}+\Delta\mathrm{t})=\mathrm{X}(\mathrm{t})+\dot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}+\frac{1}{2!}\ddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^2+\frac{1}{3!}\dddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^3+\frac{1}{4!}\dddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^4+\ldots$$

$$\mathrm{X}(\mathrm{t}-\Delta\mathrm{t})=\mathrm{X}(\mathrm{t})-\dot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}+\frac{1}{2!}\ddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^2-\frac{1}{3!}\dddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^3+\frac{1}{4!}\dddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^4+\ldots$$

+

$$\mathrm{X}(\mathrm{t}+\Delta\mathrm{t})+\mathrm{X}(\mathrm{t}-\Delta\mathrm{t})=2\mathrm{X}(\mathrm{t})+\ddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^2+O(\Delta\mathrm{t}^4)$$

or

$$\boxed{\mathrm{X}(\mathrm{t}+\Delta\mathrm{t})\approx 2\mathrm{X}(\mathrm{t})-\mathrm{X}(\mathrm{t}-\Delta\mathrm{t})+\ddot{\mathrm{X}}(\mathrm{t})\Delta\mathrm{t}^2}$$

Verlet algorithm

Verlet algorithm

- is time reversible
- does conserve volume in phase space
- (is “symplectic”)
- does not suffer from energy drift

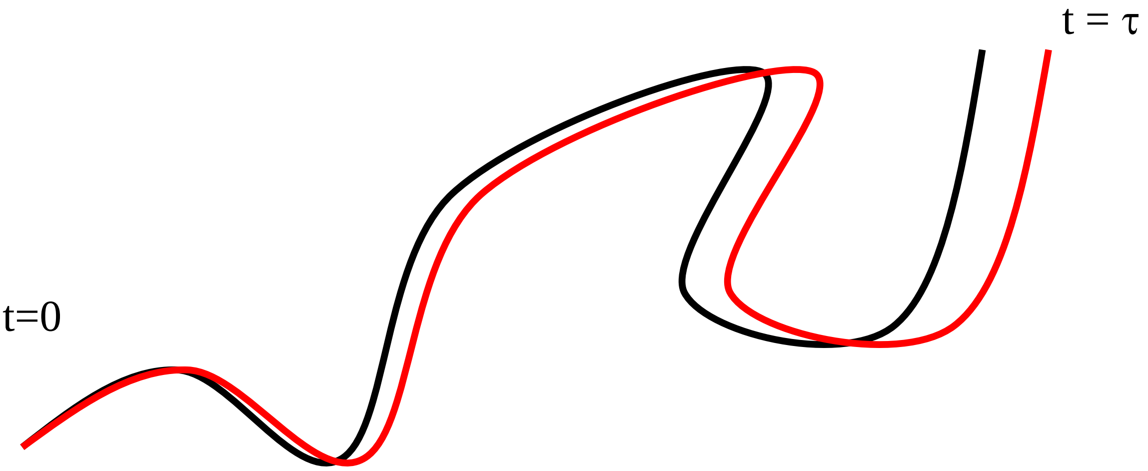
...but is it a good algorithm?

i.e. does it predict the time evolution of the system correctly???

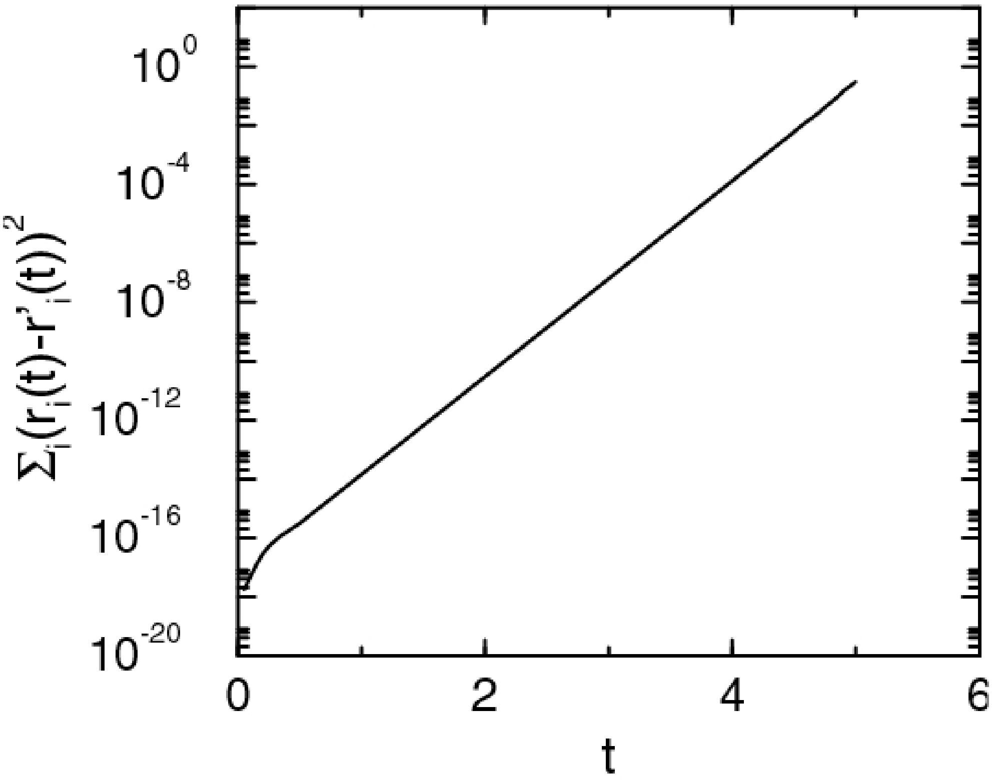
Dynamics of “well-behaved” classical many-body system is chaotic.

Consequence:

Trajectories that differ very slightly in their initial conditions DIVERGE EXPONENTIALLY (“Lyapunov instability”)



The Lyapunov disaster in action...



Any small error in the numerical integration of the equations of motion, will blow up exponentially....

always...

...and for any algorithm!!

SO:

Why should anyone believe Molecular Dynamics simulations ???

Answers:

1. In fact, one should not...

exit Molecular Dynamics...

Answers:

1. In fact, one should not...
2. Good MD algorithms (e.g. Verlet) can also be considered as good Monte Carlo algorithms – they therefore yield reliable **STATIC** properties (“Hybrid Monte Carlo”)

What is the point of simulating dynamics, if we cannot trust the resulting time-evolution???

Answers:

1. In fact, one should not...
2. Good MD algorithms (e.g. Verlet) can also be considered as good Monte Carlo algorithms – they therefore yield reliable **STATIC** properties (“Hybrid Monte Carlo”)

3. All is well (probably), because of...

The Shadow Theorem....

For any realistic many-body system, the shadow theorem is merely a hypothesis.

It states that (my words):

Good algorithms generate numerical trajectories that are “close to” a REAL trajectory of the many-body system.

Question:

Does the Verlet algorithm indeed generate “shadow” trajectories?

Take a different look at the problem.

Do not discretize NEWTON's equation of motion...

...but discretize the ACTION

Intermezzo:

Classical mechanics – the Lagrangian approach.

Newton:

$$\mathbf{F_i} = m_i \ddot{\mathbf{r}}_i$$

Lagrange:

Consider a system that is at a point $\mathbf{r_0}$ at time t=0 and at point $\mathbf{r_t}$ at time t=t, then:

The system follows a trajectory r(t) such that:

“Action”

$$S \equiv \int_0^t dt' \mathcal{L}(\mathbf{r}(t))$$ is an extremum.

“Lagrangian”

Where the Lagrangian is defined as:

$$\mathcal{L}(\mathbf{r}(t)) = T_{kinetic} - U_{pot}(\mathbf{r})$$

For example, if we use cartesian coordinates:

$$\mathcal{L}(\mathbf{r}(t)) = \sum_{i=1}^{N} \frac{1}{2} m_i \dot{\mathbf{r}}_i^2 - U(\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N)$$

What does this mean?

Consider the “true” path $\mathbf{R}(t)$, with $\mathbf{R}(0)=\mathbf{r}_0$ andR$(t)=\mathbf{r}_t$.

Now, consider a path close to the true path:

$$\mathbf{r}(t') = \mathbf{R}(t') + \delta\mathbf{r}(t')$$

Then the action **S** is an extremum if

$$\frac{\delta S}{\delta \mathbf{r}(t')} = 0 \quad , (\forall t')$$

(what does this equation mean??)

$$S_{\text{continuous}} = \int_{t_0}^{t_1} dt\, L(t)$$

Discretized version

$$S_{\text{discrete}} = \Delta t \sum_{i=0}^{i_{\max}} L(t_i)$$

$$L(t_i) = T(t_i) - U(t_i)$$

e.g. for one coordinate in one dimension

$$L(t_i)\Delta t = \frac{1}{2} m\Delta t \frac{(X_{i+1} - X_i)^2}{\Delta t^2} - U(X_i)\Delta t$$

and hence the discretized action is

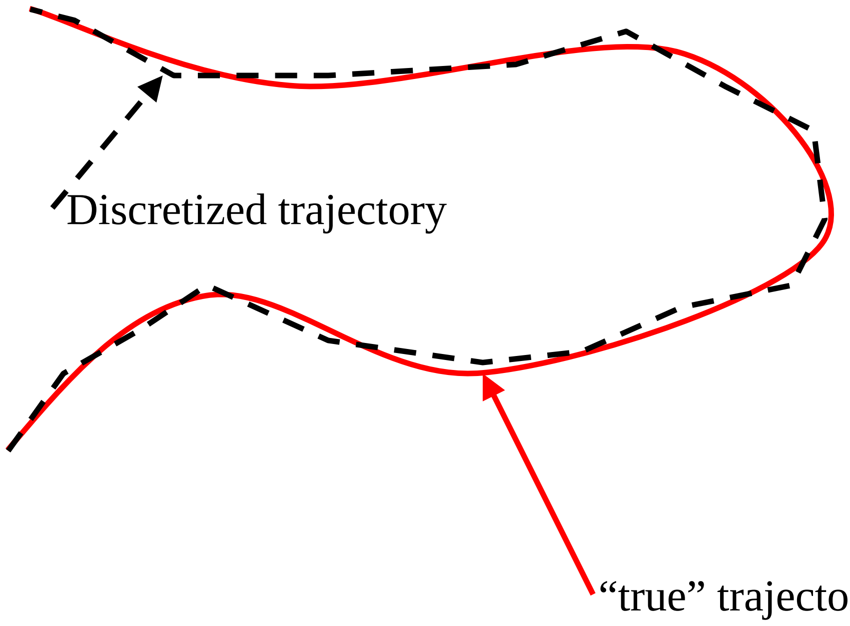
$$S_{\text{discrete}} = \sum_{i=1}^{i_{\max}} \left(\frac{m(X_{i+1} - X_i)^2}{2\Delta t} - U(X_i)\Delta t \right)$$

Now do the standard thing:

Find the extremum for small variations in the path, i.e. for small variations in all X_i.

$$\frac{\partial S_{\text{discrete}}}{\partial X_i} = 0 \qquad (\forall i)$$

This will generate a discretized trajectory that starts at time t_0 at X_0, and ends at time t_1 at X_1.



$$\frac{\partial S_{\mathrm{discrete}}}{\partial X_i} = \frac{\partial}{\partial X_i} \sum_{i=1}^{i_{\max}} \left(\frac{m(X_{i+1} - X_i)^2}{2\Delta t} - U(X_i)\Delta t \right)$$

$$\frac{\partial S_{\mathrm{discrete}}}{\partial X_i} = \frac{-m(X_{i+1} - X_i) + m(X_i - X_{i-1})}{\Delta t} - \Delta t \frac{\partial U(X_i)}{\partial X_i}$$

And hence:

$$0 = \frac{m}{\Delta t} \left(2X_i - X_{i+1} - X_{i-1} - \frac{\Delta t^2}{m} \frac{\partial U(X_i)}{\partial X_i} \right)$$

$$0 = \left(2X_i - X_{i+1} - X_{i-1} - \frac{\Delta t^2}{m} \frac{\partial U(X_i)}{\partial X_i} \right)$$

REWRITE AS:

$$X_{i+1} = 2X_i - X_{i-1} + \frac{\Delta t^2}{m} F(X_i)$$

VERLET!!!

The Verlet algorithm generates trajectory that satisfies the boundary conditions of a REAL trajectory – both at the beginning and at the endpoint.

Hence, if we are interested in statistical information about the dynamics (e.g. time-correlation functions, transport coefficients, power spectra...)

...then a “good” MD algorithm (e.g. Verlet) is fine.

ADVANCED MC SAMPLING

Conventional MC performs a RANDOM WALK in configuration space, such that

The number of times that each point is visited, is proportional to its Boltzmann weight.

$$n(\mathbf{r}^N) = c\,\mathrm{exp}[-\beta\mathcal{U}(\mathbf{r}^N)]$$

Can we achieve 100% acceptance?

Solution of conflict: play with the a-priori probabilities of trial moves:

$$\alpha(o \rightarrow n) \neq \alpha(n \rightarrow o)$$

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{\alpha(n \rightarrow o)}{\alpha(o \rightarrow n)} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}.$$

In particular, if:

$$\frac{\alpha(n \rightarrow o)}{\alpha(o \rightarrow n)} = \exp\{-\beta[\mathcal{U}(o) - \mathcal{U}(n)]\}.$$

Then

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = 1$$

(100% acceptance)

100% acceptance can be achieved in special cases: e.g. Swendsen-Wang, Wolff or Bortz-Kalos-Lebowitz…

But, often at a price…

Example: Swendsen-Wang

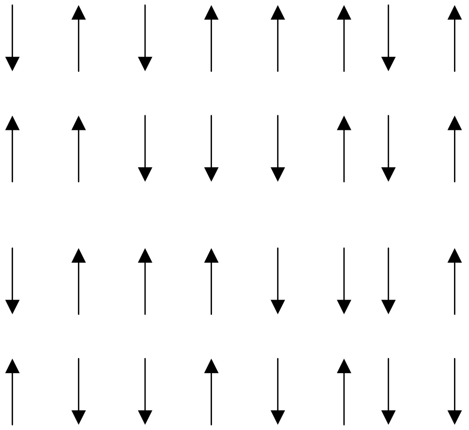
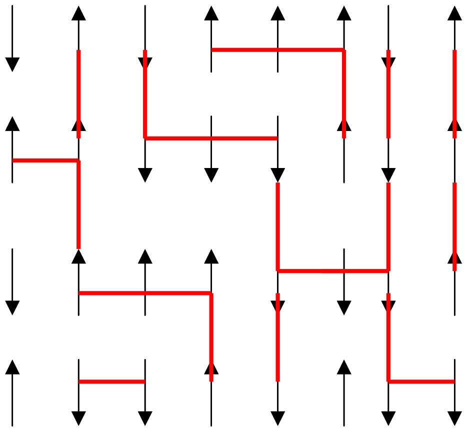


Illustration: 2D Ising model.

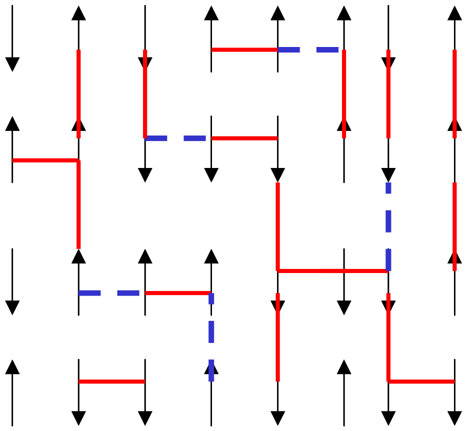
Snapshot: some neighbors are parallel, others anti-parallel



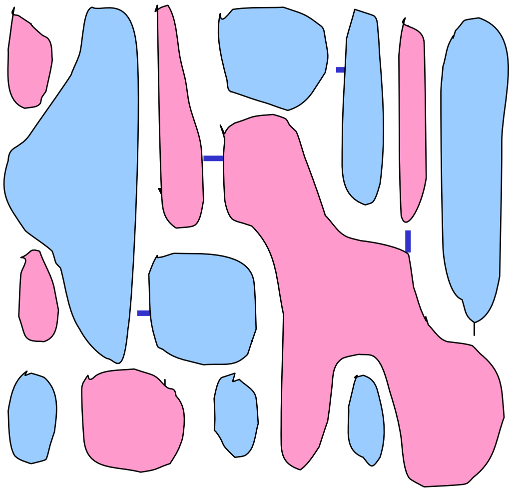
Number of parallel nearest-neighbor pairs: $\mathbf{N_p}$

Number of anti-parallel nearest neighbor pairs is: $\mathbf{N_a}$

Total energy: **$U = (N_a - N_p)\,J$**

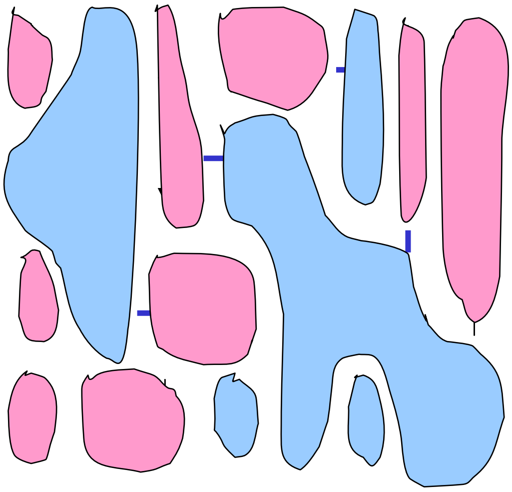


Make “bonds” between parallel neighbors. The probability to have a bond (red line) between parallel neighbors is **p** (as yet undetermined). With a probability **1-p**, parallel neighbors are not connected (blue dashed line).



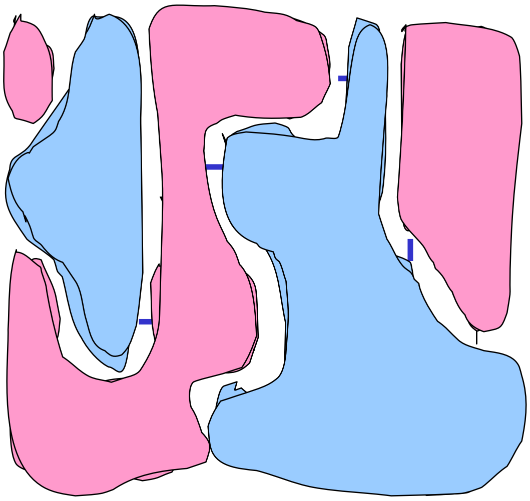
Form clusters of all spins that are connected by bonds. Some clusters are all "spin up" others are all "spin down".

Denote the number of clusters by **M**.

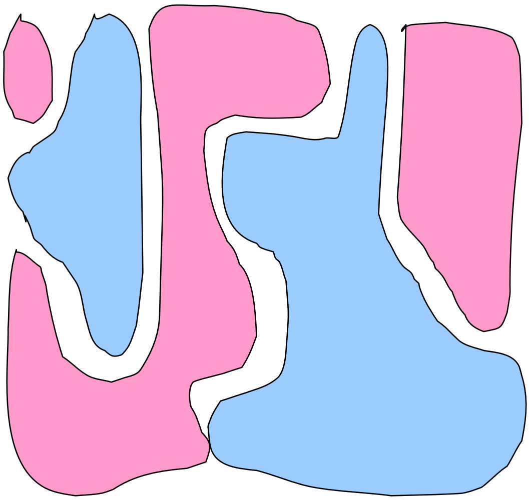


Now randomly flip clusters. This yields a new cluster configuration with probability $\mathbf{P_{(flip)} = (1/2)^M}$.

Then reconnect parallel spins



Next: forget about the “bonds”…



New spin configuration!

$$P_o P_{clus}(o) P_{flip}(M) P_{acc}(o \rightarrow n)$$

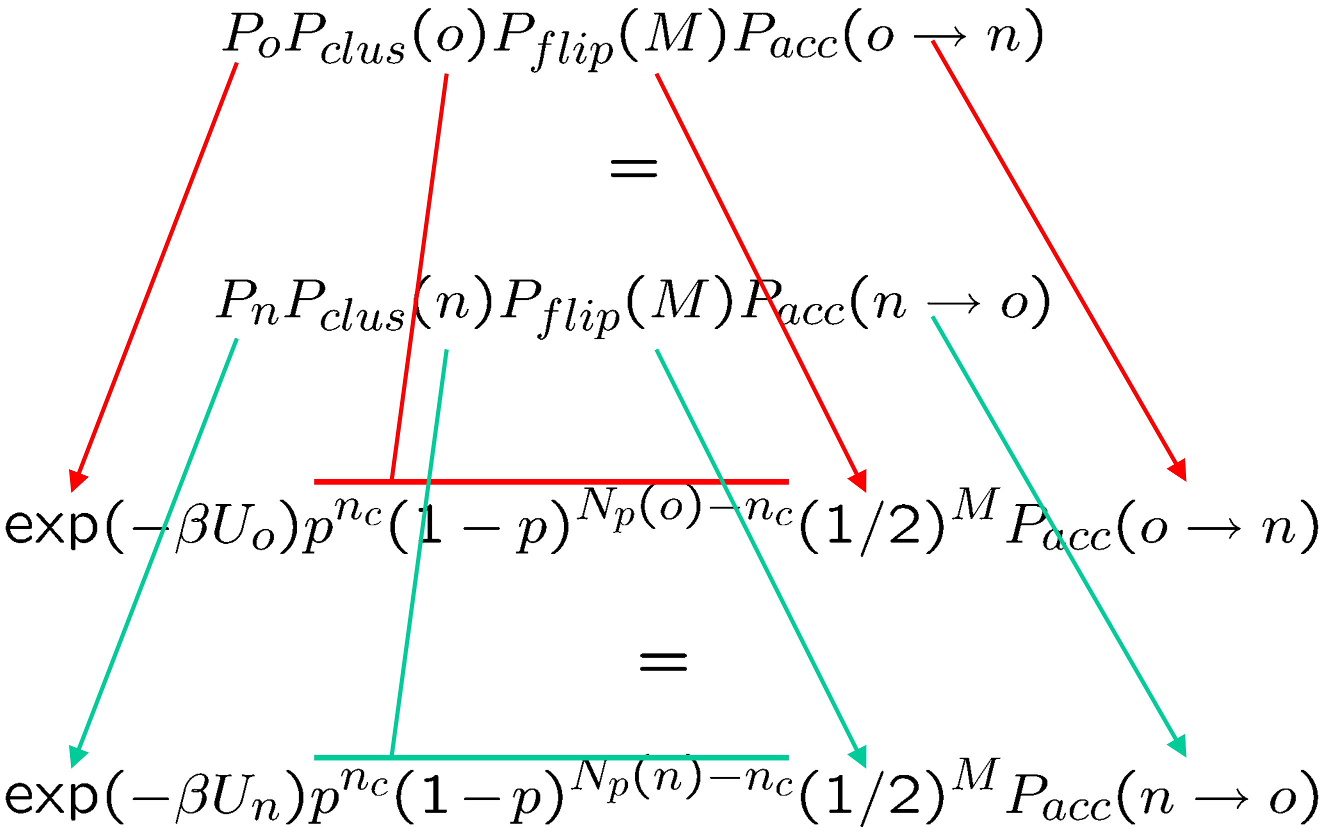
$$=$$

$$P_n P_{clus}(n) P_{flip}(M) P_{acc}(n \rightarrow o)$$

$$\exp(-\beta U_o) p^{n_c} (1-p)^{N_p(o)-n_c} (1/2)^M P_{acc}(o \rightarrow n)$$

$$=$$

$$\exp(-\beta U_n) p^{n_c} (1-p)^{N_p(n)-n_c} (1/2)^M P_{acc}(n \rightarrow o)$$



$$P_o P_{clus}(o) P_{flip}(M) P_{acc}(o \rightarrow n)$$

$$=$$

$$P_n P_{clus}(n) P_{flip}(M) P_{acc}(n \rightarrow o)$$

$$\exp(-\beta U_o) p^{n_c} (1-p)^{N_p(o)-n_c} (1/2)^M P_{acc}(o \rightarrow n)$$

$$=$$

$$\exp(-\beta U_n) p^{n_c} (1-p)^{N_p(n)-n_c} (1/2)^M P_{acc}(n \rightarrow o)$$

Moreover, we want 100% acceptance, i.e.:

$P_{acc}(o \rightarrow n) = P_{acc}(n \rightarrow o) = 1$

$$\exp(-\beta U_o)\cancel{p^{n_c}}(1-p)^{N_p(o)-\cancel{n_c}}(\cancel{1/2})^M P_{acc}(o \rightarrow n)$$

$$=$$

$$\exp(-\beta U_n)\cancel{p^{n_c}}(1-p)^{N_p(n)-\cancel{n_c}}(\cancel{1/2})^M P_{acc}(n \rightarrow o)$$

Hence:

$$\exp(-\beta U_o)(1-p)^{N_p(o)} = \exp(-\beta U_n)(1-p)^{N_p(n)}$$

$$\exp(\beta(U_n - U_o)) = (1-p)^{N_p(n)-N_p(o)}$$

But remember:

$$U_n - U_o = J(N_a(n) - N_p(n)) - J(N_a(o) - N_p(o))$$

or

$$\Delta U = J(\Delta N_a - \Delta N_p)$$

But: $\Delta N_a = -\Delta N_p$
and therefore
$\Delta U = -2J\Delta N_p$

$$\exp(\beta(U_n - U_o)) = \exp(-2\beta J(N_p(n) - N_p(o)))$$

Combining this with:

$$\exp(\beta(U_n - U_o)) = (1-p)^{N_p(n)-N_p(o)}$$

we obtain:

$$p = 1 - \exp(-2\beta J)$$

100% acceptance!!!

HERETICAL MC SAMPLING

Include “rejected” moves in the sampling

Why is this heretical?:

Metropolis “importance” sampling is **based** on the earlier (Ulam/von Neumann) **rejection** method applied to **random** MC sampling

---

This is the key:

$$\sum_m \rho(m)\pi_{mn} = \rho(n)$$

The transition matrix π leaves the equilibrium distribution ρ unchanged.

$$\langle A \rangle_\rho = \sum_n A_n \rho_n$$

This, we can rewrite as:

$$\sum_n A_n \rho_n = \sum_n \sum_m A_n \rho_m \pi_{mn} = \sum_m \rho_m \sum_n A_n \pi_{mn}$$

$$= \sum_m \rho_m \sum_n A_n \pi_{mn} \Leftrightarrow \langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

Note that <A> is no longer an average over “visited” states – we also include “rejected” moves in the sampling.

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

This relation also holds for any set of "connected" trial states: i.e. the possible final states of a single (decent) MC trial move.

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

For instance: in conventional MC, there would be only two states (the "old" state and the "new" state)

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

But in other algorithms, there are many. e.g. in the Swendsen-Wang algorithm: **n** clusters that can be flipped ⇒ $\mathbf{2^n}$ connected states.

The more parallel the algorithm, the better…

$$\langle A \rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

Note that the transition matrix that is used in the averaging need not be the same as the one used in sampling the ρ_m.

(e.g. one could be “Barker” and the other “Metropolis”)

$$\langle A\rangle_\rho = \left\langle \sum_n \pi_{mn} A_n \right\rangle_{\rho_m}$$

How to sample this?

$$\pi_{nm} = \alpha_{nm} P_{acc}(nm)$$

$$\langle A\rangle_\rho = \sum_m \rho_m \left\langle \sum_n P_{acc}(mn) A_n \right\rangle_\alpha$$

For example: Barker (symmetric) rule

$$P_{acc}(nm) = \frac{e^{-\beta U_m}}{e^{-\beta U_n} + e^{-\beta U_m}}$$

$$\langle A\rangle_\rho = \sum_m \rho_m \left\langle \frac{e^{-\beta U_n} A_n + e^{-\beta U_m} A_m}{e^{-\beta U_n} + e^{-\beta U_m}} \right\rangle_\alpha$$

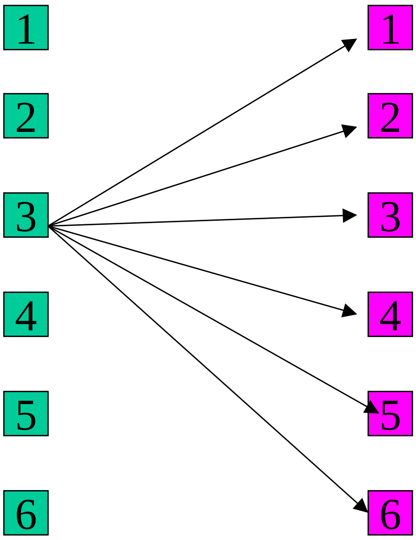
(gives exact result in one step, for two-level system)

Note: You can do many “test moves” starting from a single starting configuration m.

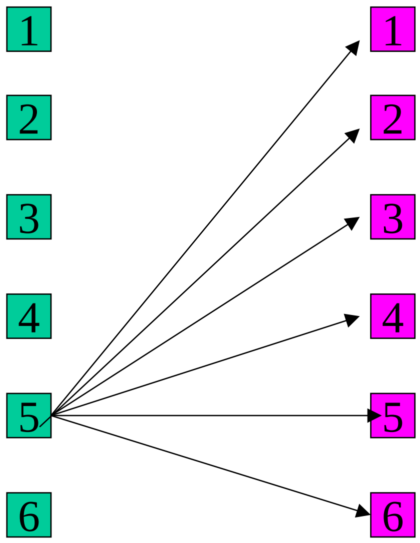
Is there any advantage in doing so?

YES, SOMETIMES:

Example: “virtual move parallel tempering”



Consider all possible “swaps” starting from any initial state i (e.g. i=3)



Then from some other initial state j (e.g. j=5), etc.

IN THIS WAY, WE CAN SAMPLE VERY MANY POTENTIAL TEMPERING MOVES AT VIRTUALLY ZERO ADDED COST.

Chain Molecules

1. Repeat: Chemical Potentials
2. Rosenbluth Sampling
3. Configurational Bias Monte Carlo
4. Other schemes

Particle insertion method to compute chemical potentials

$$\mu \equiv \left(\frac{\partial F}{\partial N}\right)_{V,T}$$

But N is not a continuous variable.
Therefore

$$\mu \approx \left(\frac{F(N+1,V,T)-F(N,V,T)}{(N+1)-N}\right)$$

$$F(N+1,V,T)-F(N,V,T) = -kT\ln\frac{Q(N+1,V,T)}{Q(N,V,T)}$$

Does that help?

Yes: rewrite

$$Q(N,V,T) = \frac{1}{N!\Lambda^{3N}}\int dr^N \exp(-\beta U(r^N)) =$$

$$\frac{1}{N!\Lambda^{3N}}V^N\int ds^N \exp(-\beta U(s^N;L)$$

s is a scaled coordinate: 0≤**s**<1

r = L **s** (is box size)

$$\frac{Q(N+1,V,T)}{Q(N,V,T)} =$$

$$\frac{V}{(N+1)\Lambda^3}\frac{\int ds^{N+1}\,\mathrm{exp}(-\beta U(s^{N+1})}{\int ds^{N}\,\mathrm{exp}(-\beta U(s^{N})}$$

Now write

$$U((s^{N+1}) \equiv U(s^N) + \Delta U(s_{N+1}, s^N)$$

then

$$\frac{Q(N+1,V,T)}{Q(N,V,T)} =$$

$$\frac{V}{(N+1)h^3}\int ds_{N+1}\langle \mathrm{exp}(-\beta \Delta U(s_{N+1}, s^N))\rangle$$

And therefore

$$\mu = -kT\ln\left(\frac{V}{(N+1)\Lambda^3}\int ds_{N+1}\langle \mathrm{exp}(-\beta \Delta U(s_{N+1}, s^N))\rangle\right)$$

$$-kT\ln\left(\frac{V}{(N+1)\Lambda^3}\right) =$$

but

$$kT\ln(\rho\Lambda^3) = \mu_{id.gas}$$

So, finally, we get:

$$\mu = \mu_{id.gas} - kT \ln \left(\int ds_{N+1} \langle \exp(-\beta \Delta U(s_{N+1}, s^N)) \rangle \right)$$

Interpretation:

1. Evaluate ΔU for a random insertion of a molecule in a system containing N molecule.
2. Compute $\exp(-\beta \Delta U)$
3. Repeat M times and compute the average "Boltzmann factor" $< \exp(-\beta \Delta U) >$
4. Then $\mu_{excess} = -kT \ln < \exp(-\beta \Delta U) >$

Particle insertion continued….

$$\mu \approx \left(\frac{F(N+1, V, T) - F(N, V, T)}{(N+1) - N} \right)$$

therefore

$$\mu \approx -kT \ln \frac{Q(N+1, V, T)}{Q(N, V, T)}$$

But also

$$\mu \approx +kT \ln \frac{Q(N, V, T)}{Q(N+1, V, T)}$$

As before:

$$Q(N,V,T) = \frac{1}{N!\Lambda^{3N}} \int dr^N \, \mathrm{exp}(-\beta U(r^N) =$$

$$\frac{1}{N!\Lambda^{3N}} V^N \int ds^N \, \mathrm{exp}(-\beta U(s^N; L)$$

With **s** a scaled coordinate: 0≤**s**<1

r = L **s** (is box size)

$$\frac{Q(N,V,T)}{Q(N+1,V,T)} =$$

$$\frac{(N+1)\Lambda^3}{V} \frac{\int ds^N \, \mathrm{exp}(-\beta U(s^N)}{\int ds^{N+1} \, \mathrm{exp}(-\beta U(s^{N+1})}$$

$$= \frac{(N+1)\Lambda^3}{V} \frac{\int ds_{N+1} \int ds^N \, \mathrm{exp}(-\beta U(s^N)}{\int ds^{N+1} \, \mathrm{exp}(-\beta U(s^{N+1})}$$

Now write

$$U((s^N) \equiv U(s^{N+1}) - \Delta U(s_{N+1}, s^N)$$

$$\frac{Q(N,V,T)}{Q(N+1,V,T)} =$$

$$\frac{(N+1)\Lambda^3}{V}\langle \mathrm{exp}(+\beta\Delta U(s_{N+1},s^N))\rangle_{N+1}$$

And therefore

$$\mu = \mu_{id.gas} + kT \ln\left(\langle \mathrm{exp}(+\beta\Delta U(s_{N+1},s^N))\rangle_{N+1}\right)$$

~~Interpretation:~~

1. ~~Evaluate ΔU for a random **REMOVAL** of a molecule in a system containing N+1 molecule.~~
2. ~~Compute $\mathrm{exp}(+\beta\Delta U)$~~
3. ~~Repeat M times and compute the average “Boltzmann factor”~~
4. ~~Then~~

~~$\mu_{excess} = +kT \ln < \mathrm{exp}(+\beta\Delta U) >$~~

What is wrong?

$\exp(+\beta\Delta U)$ is not bounded. The average that we compute can be dominated by INFINITE contributions from points that are NEVER sampled.

What to do?

Consider:

$$p_0(\Delta U) \equiv \frac{\int \exp(-\beta U_N)\delta(\Delta U - U_{N+1} + U_N)}{\int \exp(-\beta U_N)}$$

And also consider the distribution

$$p_1(\Delta U) \equiv \frac{\int \exp(-\beta U_{N+1})\delta(\Delta U - U_{N+1} + U_N)}{\int \exp(-\beta U_{N+1})}$$

p_0 and p_1 are related:

$$p_1(\Delta U) = \frac{\int \exp(-\beta(U_N + \Delta U))\delta(\Delta U - U_{N+1} + U_N)}{Q_{N+1}}$$

$$= \exp(-\beta\Delta U)\frac{\int \exp(-\beta U_N)\delta(\Delta U - U_{N+1} + U_N)}{Q_{N+1}}$$

$$= \exp(-\beta\Delta U)\frac{Q_N}{Q_{N+1}}\frac{\int \exp(-\beta U_N)\delta(\Delta U - U_{N+1} + U_N)}{Q_N}$$

so

$$p_1(\Delta U) = \exp(-\beta\Delta U)\frac{Q_N}{Q_{N+1}}p_0(\Delta U)$$

but $\frac{Q_N}{Q_{N+1}} = \exp(+\beta\mu)$ hence

$$p_1(\Delta U) = \exp(-\beta\Delta U)\exp(\beta\mu)p_0(\Delta U)$$

Rewrite:

$$\ln(p_1(\Delta U)) = -\beta\Delta U + \beta\mu + \ln(p_0(\Delta U))$$

define

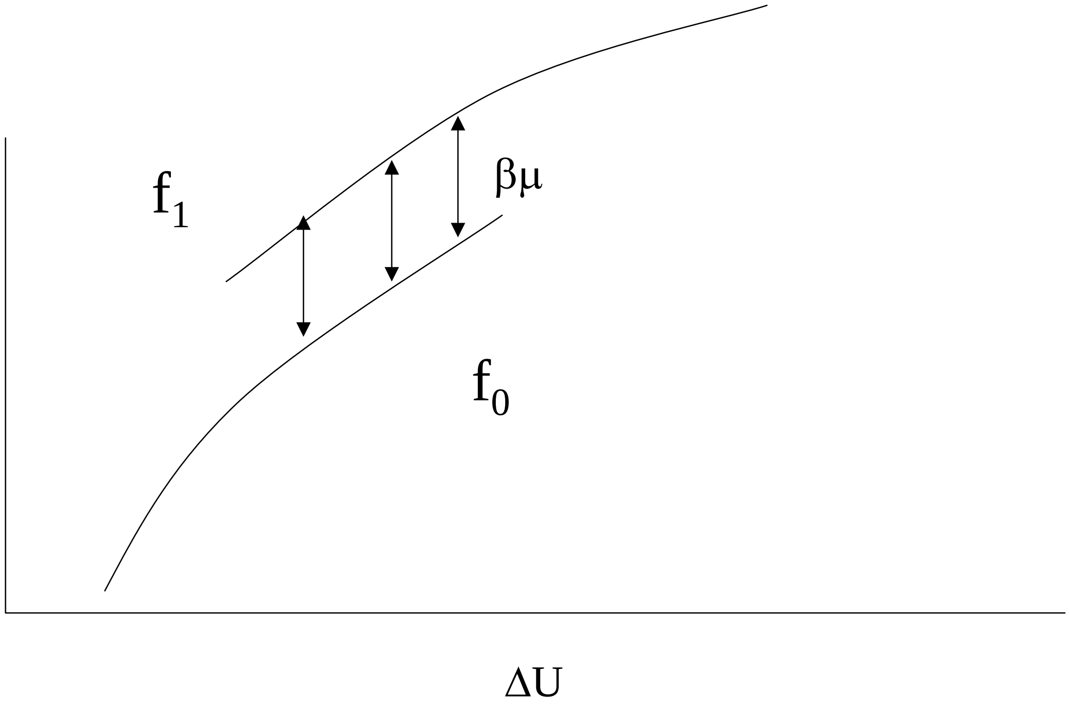
$$f_1(\Delta U) \equiv \ln(p_1(\Delta U)) + \beta\Delta U/2$$

$$f_0(\Delta U) \equiv \ln(p_0(\Delta U)) - \beta\Delta U/2$$

Then:

$$f_1(\Delta U) = f_0(\Delta U) + \beta\mu$$

For **ALL** ΔU



Diagnostic test:

If the distributions do not overlap, **don’t trust your results for μ**

(similar expression for path sampling – next lecture)

Does it work for hard spheres?

consider ΔU=0

$$f_1(0) = f_0(0) + \beta\mu$$

$$f_1(0) = \ln(1) + (constant)$$
$$f_0(0) = \ln(P_{acc}) + (constant)$$

$$\beta\mu = -\ln(P_{acc})$$

Problems with Widom method:

Low insertion probability yields poor statistics.

For instance:

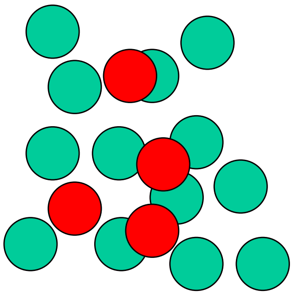
Trial insertions that consist of a sequence of intermediate steps.

Examples: changing polymer conformations, moving groups of atoms, …

What is the problem with polymer simulations?

Illlustration:

Inserting particles in a dense liquid



Trial moves that lead to "hard-core" overlaps tend to be rejected.

ANALOGY:

Finding a seat in a crowded restaurant.

Can you seat one person, please…

Next: consider the random insert
molecule (polymer).

Waiter! Can you seat 100 persons… together please!

Random insertions of polymers in dense liquids usually fail completely…

(Partial) Solution: Biased insertion.

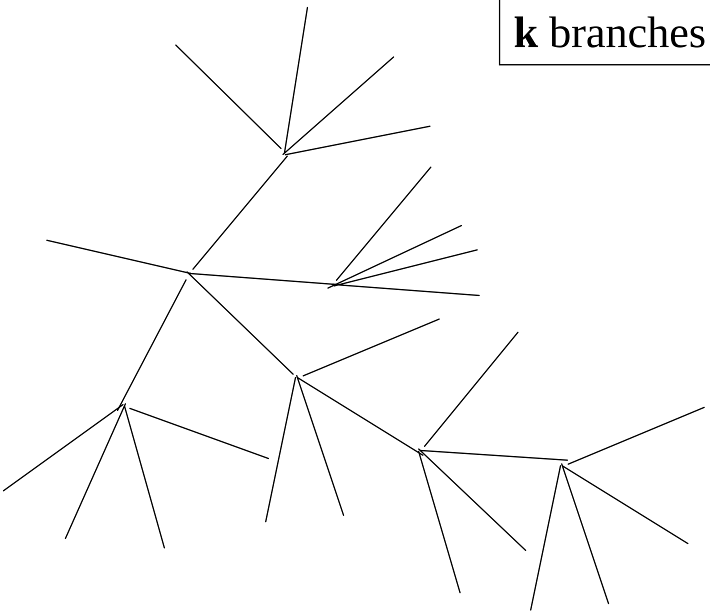
1. Chain insertion on a lattice
2. CBMC (on lattice)
3. CBMC (off lattice)

I MUST do this on the black board,

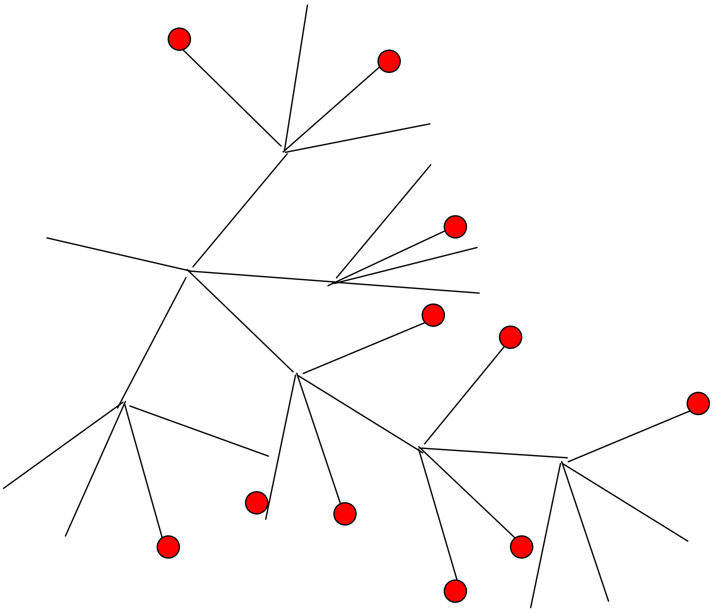
Otherwise it goes too fast…

Consider a “tree” of possible polymer conformations.

k branches start at every vertex.



But many branches terminate (due to hard-core overlaps)



Now we can compare what different algorithms do:

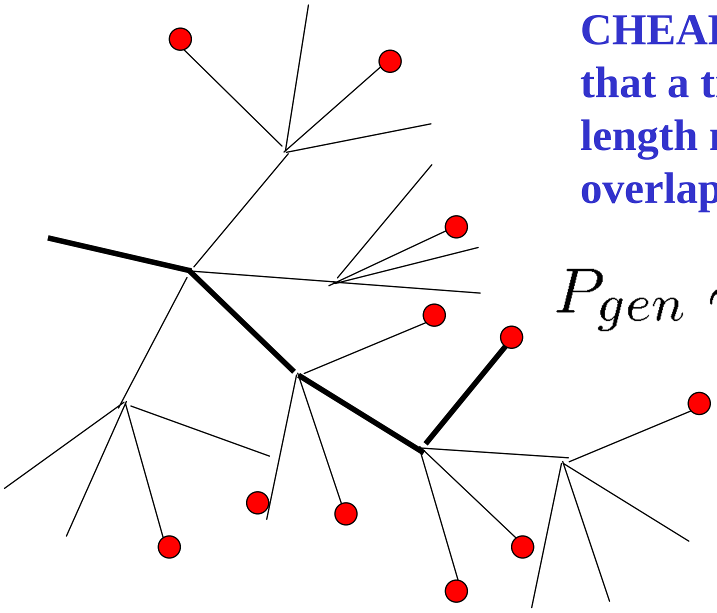
1. Conventional (unbiased) Monte Carlo
2. Configurational Bias Monte Carlo (CBMC)
3. Dynamic Pruning-Enrichment Rosenbluth Monte Carlo
4. Recoil-Growth Monte Carlo (RG)

None of these methods explores the complete tree of trial directions…

…but some, more than others.

J.I. Siepmann, J.J. de Pablo, B. Smit, M.W.Deem, G. Mooij, J.T.H.Vlugt, S.Consta, N. Wilding, N. Combe, P.R. ten Wolde, …

Conventional Monte Carlo generates 1 trial conformation



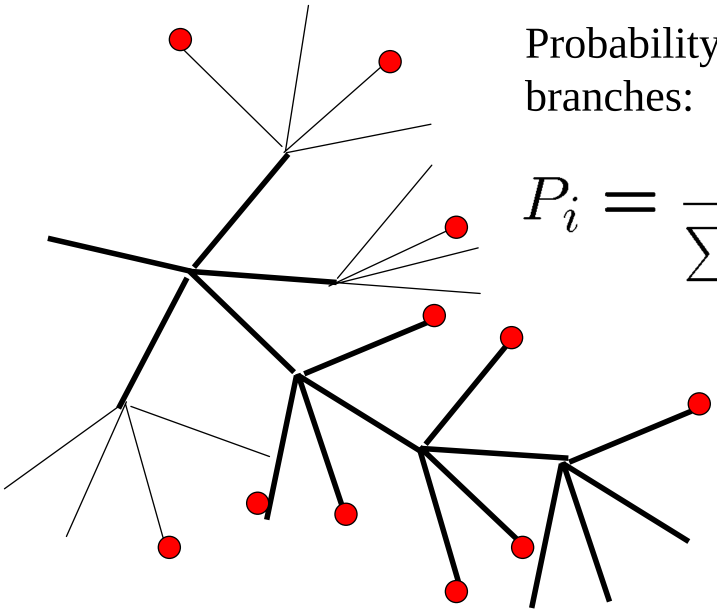
CHEAP…but the probability that a trial conformation of length n is free of hard-core overlaps, scales as

$$P_{gen} \sim (1 - p_{overlap})^n$$

Hopeless, for all but the shortest chains.

Configurational-Bias MC: an off-lattice, dynamic version of the static, on-lattice Rosenbluth-Rosenbluth scheme.

Approach: Look one step ahead.



Probability to select one of the **k** branches:

$$P_i = \frac{\exp(-\beta u_i)}{\sum_{j=1}^{k} \exp(-\beta u_j)}$$

This biases the selection of trial conformation: you never select a segment that “dies” immediately

We correct for the bias in the acceptance step

Define the “Rosenbluth weight” (**w**) of the old (new) conformations of a chain of length **n**:

$$w^{(o)} = \prod_{\ell=1}^{n} \left(\sum_{j=1}^{k} \mathsf{exp}(-\beta u^{(o)}(j;\ell)) \right)$$

The trial conformation is accepted with a probability:

$$P_{acc} = \mathsf{min}\{1, w^{(n)}/w^{(o)}\}$$

Explanation (for a 1-segment chain). Detailed balance:

$$\mathsf{exp}(-\beta U_o) \times \alpha(o \to n) \times \mathsf{acc}(o \to n)$$

$$=$$

$$\mathsf{exp}(-\beta U_n) \times \alpha(n \to o) \times \mathsf{acc}(n \to o)$$

α(o→ n) is the product of the probability to generate a set of **k** trial directions

$$P_{gen}^{(n)}(\{\mathbf{u}_k\}) = P_{gen}(\mathbf{u}_1) \cdots P_{gen}(\mathbf{u}_k)$$

…times the probability to select direction **i**

$$P_i = \frac{\mathsf{exp}(-\beta u_i)}{\sum_{j=1}^{k} \mathsf{exp}(-\beta u_j)} = \mathsf{exp}(-\beta U_n^{(ext)})/w^{(n)}$$

Detailed balance

$$\cancel{\exp(-\beta U_o)} \times$$
$$\cancel{\exp(-\beta U_n^{(ext)})}/w^{(n)} \times \cancel{P_{gen}(\mathbf{u}^{(n)})} P_{gen}^{rest}(n) \times$$
$$\mathrm{acc}(o \to n)$$

equals

$$\cancel{\exp(-\beta U_n)} \times$$
$$\cancel{\exp(-\beta U_o^{(ext)})}/w^{(o)} \times \cancel{P_{gen}(\mathbf{u}^{(o)})} P_{gen}^{rest}(o) \times$$
$$\mathrm{acc}(n \to o)$$

Many terms cancel, but not all…

$$1/w^{(n)} \times P_{gen}^{rest}(n) \times \mathrm{acc}(o \to n) =$$
$$1/w^{(o)} \times P_{gen}^{rest}(o) \times \mathrm{acc}(n \to o)$$

But now we are stuck because the acceptance ratio depends on the set of trial directions for both the forward and reverse moves:

$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{w^{(n)}}{w^{(o)}} \times \frac{P_{gen}^{rest}(o)}{P_{gen}^{rest}(n)}$$

Solution: consider the generation of trial directions around the old conformation as part of the trial move…

During a trial move, we generate a set of **k** trial directions for the **new** conformation **and** a set of **k-1** trial directions around the old conformation. Then

$$\cancel{\exp(-\beta U_o)} \times$$
$$\cancel{\exp(-\beta U_n^{(ext)})}/w^{(n)} \cancel{\times P_{gen}(\mathbf{u}^{(n)}) P_{gen}^{rest}(n) \times}$$
$$\cancel{P_{gen}^{rest}(o)} \times \mathrm{acc}(o \rightarrow n)$$

must equal

$$\cancel{\exp(-\beta U_n) \times}$$
$$\cancel{\exp(-\beta U_o^{(ext)})}/w^{(o)} \cancel{\times P_{gen}(\mathbf{u}^{(o)}) P_{gen}^{rest}(o) \times}$$
$$\cancel{P_{gen}^{rest}(n) \times} \mathrm{acc}(n \rightarrow o)$$

SUPER-DETAILED BALANCE

Or:

$$\frac{\mathrm{acc}(o \rightarrow n)}{\mathrm{acc}(n \rightarrow o)} = \frac{w^{(n)}}{w^{(o)}}$$

Note:

1. Superdetailed balance is also implicit in other algorithms (e.g. Swendsen-Wang)
2. The algorithm is valid for all **k**, but the optimal values depend on density, temperature and position in the polymer.
3. Also works for ring, star and grafted polymers.

Problems:

1. Chain generation can easily terminate in a dead alley
2. Much computational effort may be wasted in the generation of conformations with a low Rosenbluth weight

---

DPERM (Dynamic Pruning-Enrichment Rosenbluth Monte Carlo)

- based on a static MC scheme due to **Grassberger** et al. -

1. Pruning

Grow trial conformations as in CBMC. For every length **m**, compute the **partial** Rosenbluth weight w(m):

$$w(m) = w(m-1) \times \frac{\sum_{j=1}^{k} \exp(-\beta u^{(n)}(j;\ell))}{k}$$

If w(m) < w_{min} then, with 50% probability, **kill** conformation, otherwise, **double** w(m).

2. Enrichment:

   If $w(m) > w_{max}$ then: make **r** copies of the partial conformation and give each a (partial) Rosenbluth weight **w(m)/r**.

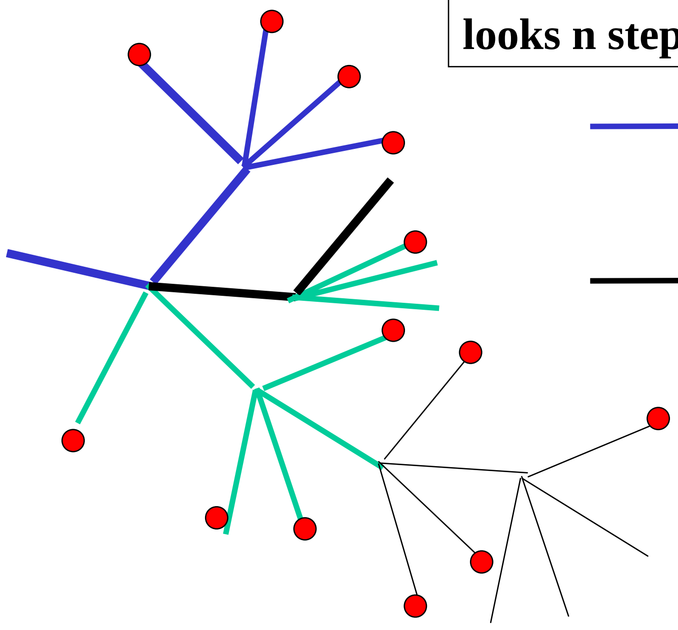
At the end, several chain conformations may survive.

The move is accepted with a probability proportional to the total Rosenbluth weight of the surviving chains.

This scheme tends to eliminate low Rosenbluth weights, and is less sensitive to dead-alleys.

Recoil-growth:

A bit like CBMC, but the scheme looks n steps ahead (here n=2).



First trial chains end in dead alley

Recoil and regrow. Success!

Finally: Check **n** steps along the unexplored conformations

For both schemes, valid acceptance rules can be constructed, based on the super-detailed balance condition.

They outperform CBMC, but mainly in the regime where all schemes are costly…

Why use such complex schemes at all?

Not always competitive for “normal” polymer simulations

- **hybrid MC**
- **reptation**
- **wormhole moves**

CBMC most useful in cases where entire chains have to be inserted (Gibbs-ensemble, Grand-Canonical MC).